```java
import java.io.*;
import java.security.*;
import javax.crypto.*;
import org.bouncycastle.jce.provider.*;


/** Clase que se puede utilizar para encriptar un fichero
 *  con una clave simétrica.
 *  La clave simetrica es encriptada con la clave publica
 *  del destinatario del mensaje.
 *  Tanto la clave simetrica encriptada
 *  como los datos encriptados con la clave simetrica son
 *  guardados en sendos ficheros
 */

public class EncriptaMensaje {

   public static void main(String[] args) {
      // Indicamos que deseamos utilizar un proveedor de algoritmos
      // de seguridad (si no se hace esto se lanza una excepcion)
      Security.addProvider(new BouncyCastleProvider());

      if (args.length==0){
         System.out.println("USO: java Encripta fichero_clave_publica
fichero_clave_encriptada fichero_datos fichero_datos_encriptado");
         System.exit(0);
      }

      // El proceso es el siguiente:
      // 1) Se genera una clave simetrica (es decir, es la misma para encriptar
      //    y desencriptar)
      // 2) Se encripta la clave simetrica generada utilizando la clave publica
      // 3) Se encriptan los datos con la clave simetrica
      // 4) Se guarda la clave simetrica encriptada con la clave publica y los
      //    datos encriptados con la clave simetrica



      try {

         // 1) Generacion de la clave simetrica

         KeyGenerator keygen = KeyGenerator.getInstance("AES","BC");
         SecureRandom random = new SecureRandom();
         keygen.init(random);
         SecretKey claveSimetrica = keygen.generateKey();

         byte[] b = claveSimetrica.getEncoded();
         int cont =0;

         for (int i=0;i<b.length;i++){
            System.out.print(b[i] + " ");
            cont = cont +1;
         }
         System.out.println("\n" + cont);


         // Lectura de la clave publica

         ObjectInputStream fichClave = new ObjectInputStream(new
FileInputStream(args[0]));
         Key clavePublica = (Key)fichClave.readObject();
         fichClave.close();

         // 2) Cifrado de la clave simetrica con la clave publica

         Cipher cifrador = Cipher.getInstance("RSA", "BC");

         cifrador.init(Cipher.WRAP_MODE, clavePublica);

         byte[] claveCifrada = cifrador.wrap(claveSimetrica);



         // 3) - 4) Se escribe en el fichero la longitud y la clave
         // simetrica cifrada

         DataOutputStream out = new DataOutputStream(new FileOutputStream(args[1]));
         System.out.println(claveCifrada.length);
         out.writeInt(claveCifrada.length);
```

```
            out.write(claveCifrada);
            out.close();


            InputStream in = new BufferedInputStream(new FileInputStream(args[2]));
            OutputStream fichCod = new BufferedOutputStream(new FileOutputStream(args[3]));


            Cipher simetrico = Cipher.getInstance("AES","BC");

            // Con Cipher.ENCRYPT_MODE se indica que se se va a utilizar
            // el objeto Cipher para encriptar datos
            simetrico.init(Cipher.ENCRYPT_MODE, claveSimetrica);


            Cifra.cifra(in,fichCod,simetrico);

            in.close();
            fichCod.close();

        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (GeneralSecurityException e) {
            e.printStackTrace();
        }

    }
}
```